

Команда и вклад

Подбор тестов и подготовка теоретического материала - Анастасия
Написание кода – Кирилл

Постановка задачи:

- 1) Реализовать функцию `nash_equilibrium(a)`, принимающую матрицу выигрышей и возвращающую оптимальные стратегии обоих игроков и цену игры.
- 2) Визуализировать спектр оптимальных стратегий с помощью Jupiter Notebook для игр с:
 - i. полным спектром оптимальных стратегий
 - ii. неполным спектром оптимальных стратегий
 - iii. спектром, состоящим из одной точки

Инструкция по запуску:

Открыть файл `final.ipynb` в Jupiter Notebook

Подход к решению задачи и что было использовано:

- 1) Для реализации `nash_equilibrium(a, test = 0)` была использована функция `linprog(c, Aq, bq)` из библиотеки SciPy. Поскольку эта ф-ция ищет минимум, то для решения прямой задачи необходимо применить хитрость: использовать ее для $(-1) \cdot a^T$ - таким образом мы будем находить не минимум, а максимум. Вид функции накладывает некоторые ограничения на ее использования. Для облегчения задания некоторых ее параметров создана функция `mini`, проверяющая матрицу выигрышей на наличие элементов ≤ 0 . Далее, если матрица неположительна, мы приводим ее сдвигом к положительному виду. Это возможно в силу Теоремы об аффинных преобразованиях, согласно которой преобразования сдвига и подобия не меняют решения игры. Цена игры в дальнейшем также преобразуется согласно теореме, чтобы соответствовать цене для исходной матрицы, для этого
- 2) Для визуализации спектров оптимальных стратегий была создана функция `allprint(result)`, на вход которой приходит вектор из стратегий первого и второго игрока и цены игры.
- 3) Для наглядной демонстрации действия программы был использован функционал Jupiter Notebook. В 4 парах Cells приводятся: пример из задания, примеры для 2.i, 2.ii, 2.iii

- 4) Также был создан пакет `modetask`, в котором раскиданы по модулям используемые функции. Для унифицированного применения пакета были использованы библиотеки `os` и `sys`, чтобы задать относительную ссылку `sys.path.append(os.path.join(sys.path[0], '..'))`. Это делается в силу того, что `python` по умолчанию ищет пакеты в текущей директории.
- 5) Были также проведены два `unittest`'а для сравнения результатов действия программы с некоторыми значениями, полученными эмпирическим путем